REMARKS

I. INTRODUCTION

In response to the Office Action dated July 12, 2005, claims 1, 16, and 31 have been amended, and claims 2, 17, and 32 have been cancelled. Claims 1, 3-16, 18-31, and 33-45 remain in the application. Entry of these amendments, and re-consideration of the application, as amended, is requested. Applicants also note that the amendments merely added the limitations from the cancelled claims to the independent claims. Accordingly, no further search and/or consideration is necessary at this time. Accordingly, Applicants respectfully request entry of the amendments.

III. PRIOR ART REJECTIONS

In paragraph (1) of the Office Action, claims 1, 3-5, 7, 10, 11, 16, 18-20, 22, 25, 26, 31, 33-35, 37, 40, and 41 were rejected under 35 U.S.C. §102(b) as being anticipated by Wallace et al., U.S. Patent No. 5,861,889 (Wallace). In paragraph (2) of the Office Action, claims 2, 17, and 32 were rejected under 35 U.S.C. §103(a) as being unpatentable over Wallace in view of Choi, U.S. Patent No. 6,639,606 (Choi). In paragraph (3) of the Office Action, claims 6, 8, 9, 21, 23, 24, 36, 38, and 39 were rejected under 35 U.S.C. §103(a) as being unpatentable over Wallace in view of Felser et al., U.S. Patent No. 6,025,849 (Felser). In paragraph (4) of the Office Action, claims 12, 27, and 42 were rejected under 35 U.S.C. §103(a) as being unpatentable over Wallace in view of Wang, U.S. Patent No. 4,701,752 (Wang). In paragraph (5) of the Office Action, claims 13, 28, and 43 were rejected under 35 U.S.C. §103(a) as being unpatentable over Wallace in view of Argiolas, U.S. Patent No. 5,956,032 (Argiolas). In paragraph (6) of the Office Action, claims 14, 29, and 44 were rejected under 35 U.S.C. §103(a) as being unpatentable over Wallace in view of Young, U.S. Patent No. 5,299,307 (Young). In paragraph (7) of the Office Action, claims 15, 30, and 45 are rejected under 35 U.S.C. §103(a) as being unpatentable over Wallace in view of Frank et al., U.S. Patent No. 5,651,107 (Frank).

Applicants respectfully traverse these rejections.

Specifically, the independent claims were rejected as follows:

> In regards to claims 1, 3-5, 7, 10, 11, 16, 18-20, 22, 25, 26, 31, 33-35, 37, 40, and 41-
> As shown in Fig. 1, Wallace et al. Illustrates a computer graphics system (20) with a display screen (22) showing a cylindrical drum (24) as well as a specialized graphics tool which gives the impression of a sphere about drum (24) (claims 1, 7, 10.b.i., 16.b.i., 22, 25(a), 31, 37, 40.a.: displaying a graphic object). It should be realized that other types of objects could be displayed on screen (22), either alone or in combination with other objects to yield a scene or the link [column 4, lines 58-68].

Object movement reference frame (26) is depicted as the sphere and its various handles that serve to displace (e.g., move or rotate) the displayed object on the screen [column 5, lines 13-19]. The "handles" of the object movement reference frames (26) include a plurality of object image handles (**claims 1, 7, 10.b.ii., 16.b.ii., 22, 25.b.ii., 31, 37, 40.b.**: displaying a button object manipulator comprised of a grip) [column 7, lines 5-6] including frame handles which facilitate the movement of sphere S relative to the displayed object (**claims 1, 7, 10.c., 16.b.iv., 22, 25.v., 31, 37, 40.c.**: directly modifying a property) [column 7, lines 31-34]. Referring to Fig. 13-3, if the mouse C has moved and is over the frame center knob handle (220) (as determined at step 306 of Fig. 3) (**claims 4, 19, 34**: object manipulator displayed on graphic object used to manipulate the object), step 308 is executed. At step 308, the interior of the frame center knob handle (or any of the different handles for that matter) changes color (e.g., is highlighted) (**claims 7, 22, 37**: color) and mouse pointer C changes to have a movement representation. If the left mouse button (35L) is clicked (**claims 1, 7, 10.b.iii., 11, 16.b.iii., 22, 25(c), 26, 31, 37, 40.c., 41**: activating button object manipulator), a drag frame center knob handle movement mode (or the associated mode with the chosen handle) is entered as shown by step 354 (**claims 10.b.i, 25.b.ii.1., 40.b.i.**) [column 8, lines 34-47]. If the mouse pointed is over one of the object plane handles (214), step 362 is executed to initiate object planar drag movement mode (**claims 10.b.ii., 25.b.ii.2., 40.b.ii.**) [column 9, lines 1-12]. In either case, the user then clicks the left mouse button (35L) to enter the specific mode to alter the property of the graphical object (**claims 10.d., 25.iv., 40.d.**). Thus, a discrete number of options are provided to the user to choose the desired property for manipulation of the graphical object, depending on the handle chosen (**claims 5, 20, 35**). As an example, the center knob handle movement mode will be discussed. Referring to Fig. 4, at step 402, at step 402, a determination is made whether the pointer C is being dragged (that is, whether either the mouse left button (35L) or the mouse right button (35R) is being held down). If the pointer C is being dragged, and is not over the anchor (25), then a "hit test" is performed to determine whether pointer C has the same coordinates as any one of the predetermined set of items and features of items displayed on the screen [column 10, lines 37-50]. If pointer C, while grasping frame center knob handle (220), is moved and is over a "hit" item, at step 414 a hit point in space of the displayed object being positioned is calculated. Then, at step 416, the entire object movement reference frame (26) is moved so that its frame center knob handle (220) is at the calculated (new) hit point (step 416) (**claims 4, 19, 34**: modification of the property, **claims 3, 18, 33**: graphically displayed property) [column 10, lines 51-56].

Screen (22) is generated by a display device such as a computer graphics monitor (30). The central processing unit (CPU) (32) communicates with various other constituent members of system (20) over a master bus (33) (**claims 16.a., 22, 25.a.**: computer). A user communicates with system (20) through the instrumentality of a mouse (34) and, when desired, a keyboard (36) [column 5, lines 28-35]. System (20) also has various forms of electronic memory (**claims 16.a., 22, 25.a.**: computer having memory), such as on-board random access memory (RAM) (42) and a disk (44) [column 5, lines 41-42]. In the course of operation, CPU (32) executes instructions, i.e., computer programs or computer code (**claim 31, 37, 40**: executable instructions) [column 5, lines 55-56]. Fig. 2 graphically depicts a main graphics system (MGS) (90) that is executed by CPU (32). Fig. 13-1 depicts the appearance of monitor screen (22) resulting at a certain stage if execution of MGS (90) (**claims 16, 22, 25.b.**: application executing) [column 5, lines 56-60].

Wallace et al. Discloses the limitations of **claims 6, 8, 9, 21, 23, 24, 36, 38, 39** except teaching displaying the object manipulators in viewable orientation and meaningful locations. However, Felser et al. Discloses a framework that enables the creation and maintenance of relationships between properties of objects.

As shown in Fig. 1, Wallace et al. illustrates a computer graphics system (20) with a display screen (22) showing a cylindrical drum (24) as well as a specialized graphics tool which gives the impression of a sphere about drum (24) (**claims 8, 23, 38**: displaying a graphic object). It should be realized that other types of objects could be displayed on screen (22), either alone or in combination with other objects to yield a scene or the link [column 4, lines 58-68]. Object movement reference frame (26) is depicted as the sphere and its various handles that serve to displace (e.g., move or rotate) the displayed object on the screen [column 5, lines 13-19]. The "handles" of the object movement reference frame (26) include a plurality of object image handles (**claims 8, 23, 38**: displaying a button

object manipulator comprised of a grip) [column 7, lines 5-6], including frame handles which facilitate the movement of sphere S relative to the displayed object (claims 8, 23, 38: directly modifying a property) [column 7, lines 31-34]. Referring to Fig. 13-3, if the mouse C has moved and is over the frame center knob handle (220)(as determined at step 306 of Fig. 3), step 308 is executed. At step 308, the interior of the frame center knob handle (or any of the different handles for that matter) changes color (e.g., is highlighted) and mouse printer C changes to have a movement representation. If the left mouse button (35L) is clicked (claims 8, 23, 38: activating button object manipulator), a drag frame center knob handle movement mode (or the associated mode with the chosen handle) is entered as shown by step 354 [column 8, lines 34-37]. Screen (22) is generated by a display device such as a computer graphics monitor (30). The central processing unit (CPU) (32) communicates with various other constituent members of system (20) over a master bus (33) (claims 23: computer). A user communicates with system (20) through the instrumentality of a mouse (34) and, when desired, a keyboard (36) {column 5, lines 28-35]. System (20) also has various forms of electronic memory (claims 23: computer having memory), such as on-board random access memory (RAM) (42) and a disk (44) [column 5, lines 41-42]. In the course of operation, CPU (32) executes instructions, i.e., computer programs or computer code (claim 38: executable instructions) [column 5, lines 55-56]. Fig. 2 grahically depicts a main graphics system (MGS) (90) that is executed by CPU (32). Fig. 13-1 depicts the appearance of monitor screen (22) resulting at a certain stage if execution of MGS (90) (claims 23: application executing) [column 5, lines 56-60].

Felser et al. Discloses a software system that enables the creation and maintenance of relationship between properties of objects, wherein the object can be authored by a user. It is typically implemented using a personal computer (100) as shown in Fig. 1 [column 3, line 33]. The system of Felser et al. is usually implemented in one or more application programs (118) that operate under control of the operating system (116). The application program (118) is usually a CAD program or other graphics program [column 3, lines 47-54]. Fig. 2 is a block diagram that illustrates the components of an object (200). It is comprised of a number of different elements, such as zero or more handles (210) and a drag handler (214) [column 4, lines 22-41]. A shape object (200) uses a draw objects collection (204) to define the geometry for rendering the shape object's (200) appearance [column 6, lines 8-11]. The handles (210) are points located within the shape object (200) that are exposed to the user interface when the shape object (200) is selected. Handles (210) allow direct manipulation of geometry within the shape object (200), as well as any other shape object (200) parameter of collection element that can be referenced via expressions [column 6, lines 33-37]. The handle (212) position is thus independent of mouse position, and the shape author relates the handle position (212) to the mouse pointing device (112) position by using an expression to achieve any desired handle (212) motion [column 6, lines 54-59], such as moving to a new location on the attached object when the handle (210) is being obstructed by another object (claims 6, 21, 36) or reorienting the handles (210) if they are clearly viewable (claims 8, 23, 38). Furthermore, pertaining to claims 9, 24, 39, it would have been obvious to one of ordinary skill in the art to permit multiple iterations of reorienting the handles (210) when the handles (210) are not clearly viewable so that the user will constantly be able to access the handles (210) regardless of the altered orientation.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the method and computer instructions of the movement of the handle (210) of Felser et al. within the computer programs or computer code executed by the CPU of Wallace et al. because if the graphical object displayed on the display of Wallace is rotated and/or translated so that the handles are hidden or skewed., the user would need a way to be able to be access these button object manipulators to manipulate the graphical object. Thus, the movement of the handles of Felser et al. would provide a means to move the manipulators to a visible location (claims 6, 21, 36), or reorienting them (claim 8, 23, 38) so that the user can employ them.

Wallace et al. teaches the limitation of claims 13, 28, 43 except disclosing an error condition. However Argiolas teaches a method for resizing a window that visually indicates to the user that an illegal resizing action is attempted.

As shown in Fig. 1, Wallace et al. illustrates a computer graphics system (20) with a display screen (22) showing a cylindrical drum (24) as well as a specialized graphics tool which gives the impression of a sphere about drum (24) (claims 13, 28.b.i., 43: displaying a graphic object). It should

be realized that other types of objects could be displayed on screen (22), either alone or in combination with other objects to yield a scene or the link [column 4, lines 58-68]. Object movement reference frame (26) is depicted as the sphere and its various handles that serve to displace (e.g., move or rotate) the displayed object on the screen [column 5, lines 13-19]. The "handles" of the object movement reference frame (26) include a plurality of object image handles (claims **13, 28.b.ii., 43**: displaying a button object manipulator comprised of a grip) [column 7, lines 5-6], including frame handles which facilitate the movement of sphere S relative to the displayed object [column 7, lines 31-34]. Referring to Fig. 13-3, if the mouse C has moved and is over the frame center knob handle (220) (as determined at step 306 of Fig. 3), step 308 is executed. At step 308, the interior of the frame center knob handle (or any of the different handles for that matter) changes color (e.g., is highlighted) and mouse pointer C changes to have a movement representation. If the left mouse button (35L) is clicked (claims **13, 28.b.iii., 43**: activating button object manipulator), a drag frame center knob handle movement mode (or the associated mode with the chosen handle) is entered as shown by step 354 [column 8, lines 34-47]. If the mouse pointed is over one of the object plane handles (214), step 362 is executed to initiate object planar drag movement mode (claims [column 9, lines 1-12]. In either case, the user then clicks the left mouse button (35L) to enter the specific mode to alter the property of the graphical object. Screen (22) is generated by a display device such as a computer graphics monitor (30). The central processing unit (CPU) (32) communicates with various other constituent members of system (20) over a master bus (33) (claims **28.a.**: computer). A user communicates with system (20) through the instrumentality of a mouse (34) and, when desired, a keyboard (36) [column 5, lines 28-35]. System (20) also has various forms of electronic memory (claims **28.a.**: computer having memory), such as on-board random access memory (RAM) (42) and a disk (44) [column 5, lines 41-42]. In the course of operation, CPU (32) executes instructions, i.e., computer programs or computer code (claim **43**: executable instructions) [column 5, lines 55-56]. Fig. 2 graphically depicts a main graphics system (MGS) (90) that is executed by CPU (32). Fig. 13-1 depicts the appearance of monitor screen (22) resulting at a certain stage if execution of MGS (90) (claims **28.b.**: application executing) [column 5, lines 56-60].

Argiolas teaches a computer (100) as shown in Fig. 1, comprised of a display unit (110) and a keyboard (120), as well as a processor system unit (130) which may serve to mount a fixed disk drive and a diskette drive in addition to the main processor and memory [column 3, lines 11-17]. The computer (100) preferably includes a graphic pointing device, such as a mouse (140), which may be utilized to manipulate the position of a pointer within a visual display screen (110) [column 3, lines 21-24]. The user can change the size of a window. This action is performed by acting on the cursor, which is controlled by a pointing device such as a mouse. The cursor is placed on a point of the window border (either a corner or a side) and the border is "hooked" to the cursor, so that by moving the cursor through the pointing device, the border is "dragged" and the window changes its size accordingly [column 3, 62-64]. A visual signal is provided to the user that indicates the window has reached a maximum or minimum size limit on the direction the resizing is attempted (claims **13, 28.b.iv., 43**: displaying a bitmap image at a cursor position, error condition) [column 3, lines 65-68]. Fig. 3 shows examples of such visual feedback. The hooking action of the cursor to a point on the window border would be associated with a grip on the window border.

Therefor, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include the visual error condition of Argiolas with the cursor of Wallace et al. because the visual feedbacks allow the user to immediately and intuitively perceive that an impossible action is attempted and to avoid useless and time consuming further attempts [Argiolas: column 4, lines 34-37].

Wallace et al. teaches the limitations of claims **14, 29, 44** except disclosing particular glyph shapes indicating an alignment of the graphic object with respect to one or more addition objects. However, Young discloses a computer aided design and drawing system that includes a graphic guide used for associating edges and points of one graphic image with one or more other images.

As shown in Fig. 1, Wallace et al. illustrates a computer graphics system (20) with a display screen (22) showing a cylindrical drum (24) as well as a specialized graphics tool which gives the impression of a sphere about drum (24) (claims **14, 29.b.i., 44**: displaying a graphic object). It should be realized that other types of objects could be displayed on screen (22), either alone or in

combination with other objects to yield a scene or the link [column 4, lines 58-68]. Object movement reference frame (26) is depicted as the sphere and its various handles that serve to displace (e.g., move or rotate) the displayed object on the screen [column 5, lines 13-19]. The "handles" of the object movement reference frame (26) include a plurality of object image handles (claims 14, 29.b.ii., 44: displaying a button object manipulator comprised of a grip) [column 7, lines 5-6], including frame handles which facilitate the movement of sphere S relative to the displayed object [column 7, lines 31-34]. Referring to Fig. 13-3, if the mouse C has moved and is over the frame center knob handle (220) (as determined at step 306 of Fig. 3), step 308 is executed. At step 308, the interior of the frame center knob handle (or any of the different handles for that matter) changes color (e.g., is highlighted) and mouse pointer C changes to have a movement representation. If the left mouse button (35L) is clicked, a drag frame center knob handle movement mode (or the associated mode with the chosen handle) is entered as shown by step 354 [column 8, lines 34-47]. If the mouse pointed is over one of the object plane handles (214), step 362 is executed to initiate object planar drag movement mode (claims [column 9, lines 1-12]. In either case, the user then clicks the left mouse button (35L) to enter the specific mode to alter the property of the graphical object. Screen (22) is generated by a display device such as a computer graphics monitor (30). The central processing unit (CPU) (32) communicates with various other constituent members of system (20) over a master bus (33) (claims 29.a.: computer). A user communicates with system (20) through the instrumentality of a mouse (34) and, when desired, a keyboard (36) [column 5, lines 28-35]. System (20) also has various forms of electronic memory (claims 29.a.: computer having memory), such as on-board random access memory (RAM) (42) and a disk (44) [column 5, lines 41-42]. In the course of operation, CPU (32) executes instructions, i.e., computer programs or computer code (claim 44: executable instructions) [column 5, lines 55-56]. Fig. 2 graphically depicts a main graphics system (MGS) (90) that is executed by CPU (32). Fig. 13-1 depicts the appearance of monitor screen (22) resulting at a certain stage if execution of MGS (90) (claims 29.b.: application executing) [column 5, lines 56-60].

Young teaches a computer-aided design and drawing system used for generating and manipulating graphic images on a computer display screen [column 3, lines 27-29]. Referring to Fig. 2, the geometric shapes (201), (202), and (203) are independent graphical images drawn on the display screen [column 4, lines 60-64]. These graphical images can have guide points associated with the image. For example, the guide points associated with a rectangular image comprise the four corners and the center of the rectangle [column 5, lines 23-29]. These guide points would be then handles of Wallace et al. As cursor (200) is moved from its position in Fig. 2 to its position in Fig. 3, cursor (200) enters a proximity region associated with point (211). When this occurs, guideline (210) is displayed intersecting point (211) and extending to the edges of the window (251) in which point (211) is displayed. In addition, a small circle is displayed around point (211) indicating that point (211) is the guide point associated with the guideline (210) [column 6, lines 27-37].

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include these guidelines of Young while the graphical objects of Wallace et al. are being manipulated because this would provide control for drawing and manipulating images [Young: column 1, lines 15-20] especially when dimensions are not provided [Young: column 1, lines 34-35].

Wallace et al. teaches the limitations of claims 15, 30, 45 except disclosing displaying the object manipulator in a translucent color. However, Frank et al. teaches underlying windows to display data visible to the user through window that are overlaid above an underlying window.

As shown in Fig. 1, Wallace et al. illustrates a computer graphics system (20) with a display screen (22) showing a cylindrical drum (24) as well as a specialized graphics tool which gives the impression of a sphere about drum (24) (claims 15, 30.b.i., 45: displaying a graphic object). It should be realized that other types of objects could be displayed on screen (22), either alone or in combination with other objects to yield a scene or the link [column 4, lines 58-68]. Object movement reference frame (26) is depicted as the sphere and its various handles that serves to displace (e.g., move or rotate) the displayed object on the screen [column 5, lines 13-19]. The "handles" of the object movement reference frame (26) include a plurality of object image handles (claims 15, 30.b.i., 45: displaying a button object manipulator comprised of a grip) [column 7, lines 5-6], including frame handles which facilitate the movement of sphere S relative to the displayed object [column 7, lines 31-

34]. Referring to Fig. 13-3, if the mouse C has moved and is over the frame center knob handle (220) (as determined at step 306 of Fig. 3), step 308 is executed. At step 308, the interior of the frame center knob handle (or any of the different handles for that matter) changes color (e.g., is highlighted) (claims **15, 30.b.ii., 45**: color) and mouse pointer C changes to have a movement representation. If the left mouse button (35L) is clicked, a drag frame center knob handle movement mode (or the associated mode with the chosen handle) is entered as shown by step 354 [column 8, lines 34-47]. If the mouse pointed is over one of the object plane handles (214), step 362 is executed to initiate object planar drag movement mode [column 9, lines 1-12]. In either case, the user then clicks the left mouse button (35L) to enter the specific mode to alter the property of the graphical object. Screen (22) is generated by a display device such as a computer graphics monitor (30). The central processing unit (CPU) (32) communicates with various other constituent members of system (20) over a master bus (33) (claims **30.a.**: computer). A user communicates with system (20) through the instrumentality of a mouse (34) and, when desired, a keyboard (36) [column 5, lines 28-35]. System (20) also has various forms of electronic memory (claims **30.a.**: computer having memory), such as on-board random access memory (RAM) (42) and a disk (44) [column 5, lines 41-42]. In the course of operation, CPU (32) executes instructions, i.e., computer programs or computer code (claim **45**: executable instructions) [column 5, lines 55-56]. Fig. 2 graphically depicts a main graphics system (MGS) (90) that is executed by CPU (32). Fig. 13-1 depicts the appearance of monitor screen (22) resulting at a certain stage if execution of MGS (90) (claims **30.b.**: application executing) [column 5, lines 56-60].

Frank et al. discloses a CPU coupled to a display for displaying graphic and other data in multiple overlapping windows. Referring to Fig. 10, windows 255 and 260 are displayed in display 250. Window selection buttons 280, 281, 282, and 283 are shown on the four corners of window 255. Similarly, buttons 285, 286, 287, and 288 are shown on window 260. These buttons may be thought of as object manipulators. As shown, the windows are transparent, allowing a user to see the graphic object behind a button [column 9, lines 60-65; column 10, lines 1-25].

Therefore, it would have been obvious to one of ordinary skill in the art to make also include a translucent color in the coloring of the handles of Wallace et al. as performed by Frank et al. because this would prevent the user from having an obstructed view of the entire graphical image.

Independent claims 1, 7, 8, 10, 13-16, 22, 23, 25, 28, 29-31, 37, 38, 40, and 43-45, and are generally directed to the use of button object manipulators in a computer graphics drawing program. In this regard, the graphics objects comprise one or more graphical elements. Further the button object manipulators in all of the independent claims are made up of a grip. Thus, the independent claims provide for a use in a drawing program environment with graphical elements in the drawing program. Each of the independent claims provides for different methods/mechanisms for using the button object manipulators in the drawing program.

Claims 1, 16, and 31 provide for using the button object manipulator to directly modify a property of the graphic object.

Claims 7, 22, and 37 provide for the use of colors on the object manipulator that indicates whether activation of the manipulator will affect properties of another object.

Claims 8, 23, and 38 provide for reorienting an object manipulator when the initial orientation of the manipulator is visually confusing or indistinct.

Claims 10, 25, and 40 provide for different function states of an object manipulator. Each of the function states enable the object manipulator to perform different discrete functions that modify properties of a graphic object.

Claims 13, 28, and 43 provide for displaying a bitmap image at a cursor position if the cursor position will result in an error when interacting with the object manipulator.

Claims 14, 29, and 44 provide that a particular glyph shape of the object manipulator indicates an alignment of the object with respect to additional objects.

Claims 15, 30, and 45 provide that the object manipulator is displayed in a translucent color such that the graphic object is visible behind the object manipulator.

A. <u>Independent Claims 1, 16, and 31 are Patentable Over Wallace and Choi</u>

Claims 1, 16, and 31 have been amended to include the limitations from claims 2, 17, and 32. Accordingly, the amended claims provide the ability to activate an object manipulator that is displayed on an object wherein the activation modifies a property of the graphic object. Further, the activation is performed without moving or dragging the object manipulator.

In rejecting prior dependent claims 2, 17, and 32, the Office Action relied on the combination of Wallace and Choi. In this regard, the Office Action admitted that Wallace failed to teach the activation of an object manipulator without dragging the grip/button object manipulator.

Applicants first note that the object manipulator in the claims is a button object manipulator. In other words, the object manipulator is a button. Such a teaching is contrary to that set forth in Wallace which merely provides a handle that does not appear nor function as a button whatsoever. Accordingly, the mere use of the term "button" to define the object manipulator differentiates Wallace from the present claims.

Secondly, Applicants note that Choi is not related to a drawing program whatsoever. Instead, Choi is directed towards a screen in a computer system and the ability to split the screen (see title and Abstract of Choi). The Office Action asserts that Choi's split icon, minimize icon, restore icon, and left/right split screen setting icon are equivalent to the object manipulators as claimed. Applicants respectfully disagree and traverse such an assertion. Choi's icons are exclusively used to manipulate a window. In this regard, Choi's icons do not even remotely suggest, implicitly or explicitly, the ability to manipulate or perform actions on a graphic object in a drawing program.

Simply, there is not even a remote equivalency between a window displayed on a screen and a graphic object in a drawing program. To assert that icons used to manipulate a window are the same or teach button object manipulators used to manipulate a graphic object in a drawing program is wholly without merit.

Further, Applicants submit that a person of ordinary skill in the art would not even attempt to use Choi's icons on such drawing program graphic objects. Nowhere is there any suggestion, or motivation in Wallace or Choi to use Choi's icons on such drawing program graphic objects. In this regard, MPEP §706.02(j) provides that "there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings." There is no motivation to combine Choi with Wallace in the manner suggested. The motivation provided in the Office Action is that the user will save time by not having to manipulate the object manually by dragging. While such a benefit may exist, such a motivation relies on impermissible hindsight. Under MPEP §2141.01, "The references must be viewed without the benefit of impermissible hindsight vision afforded by the claimed invention". Applicants submit that without the teaching of the present invention, one would not acknowledge the benefit asserted in the Office Action. Accordingly, the Action relies on impermissible hindsight.

In view of the above, Applicants submit that neither Wallace nor Choi, individually or combined, teach, disclose, or suggest the invention as claimed. Accordingly, Applicants respectfully request allowance of these independent claims.

B. Independent Claims 7, 22, and 37 are Patentable Over Wallace

As described above, these claims set forth specific limitations in that an object manipulator is displayed on a graphic object. In addition, a color of the object manipulator indicates whether the activation of the object manipulator will affect a property on another object. In other words, rather than the color indicating if the activation of the object will affect a property of an object that the manipulator is displayed on, the color indicates if the activation will affect a property of another object.

In rejecting these claims, the Office Action relies exclusively on Wallace. Specifically, the Action recites how at step 308 of FIG. 3 that the interior of the frame center knob handle changes

color and the mouse pointer changes to have a move representation. However, while the color of Wallace's frame center knob handle may change, the color/highlighting merely indicates that the handle on Wallace can be used (see col. 8, lines 35-40). In this regard, Wallace's highlighting/color indicates that the center knob handle can be moved to manipulate the object. However, the highlighting/color change does not even remotely reflect whether a property of another object will be affected by the activation of the handle. In this regard, the handle is merely used to manipulate the object itself and does not relate to properties of any other objects. Further, there is no suggestion, implicit or explicit, that the center knob handle or color of the handle teaches, discloses, suggests, or alludes to properties of objects other than the object on which Wallace's handle is displayed.

In view of the above, Applicants submit that Wallace's color change and highlighting merely reflects the ability to manipulate the object using the handle that changes color. Not only is the handle not equivalent to the claimed object manipulator, but the color of the claimed object manipulator reflects whether properties of an object (other than the object on which the manipulator is displayed) will be affected by activation of the manipulator. Such a teaching is not implicitly or explicitly described by Wallace. Accordingly, Applicants respectfully request allowance of the claims.

C. Independent Claims 8, 23, and 38 are Patentable Over Wallace and Felser

These independent claims relate to the orientation of an object manipulator. Specifically, if the initial orientation of the manipulator is visually confusing or indistinct, the manipulator is reoriented.

In rejecting these claims, the Office Action essentially relies on Felser col. 6, lines 54-59 and states that the handle position is thus independent of mouse position, and the shape author relates the handle position to the mouse pointing device position by using an expression to achieve any desired handle motion. Col. 6, lines 32-59 relate to handles and how the movement of a handle can be constrained. In this regard, col. 6, lines 52-58 provides:

> The shape author can use any expression, e.g., trigonometric functions, equations, or other functions to constrain handle properties. The handle 212 position is thus independent of mouse position, and the shape author relates the handle 212 position to the mouse pointing device 112 position by using an expression to achieve any desired handle 212 motion.

As can be seen from this text, the handle position may be independent of mouse position and can be related to the mouse position by an expression to achieve a particular desired handle motion. The Office Action stretches the above citation to state that desired handle motions may include moving to a new location on the attached object when the handle is being obstructed by another object or reorienting the handles if they are clearly viewable. There is not even a remote reference to moving Felser's handle to a new location if it is obstructed or reorienting the handle. While Felser provides the ability to achieve any desired handle motion, the specifically claimed ability to reorient an object manipulator is not disclosed, taught, or suggested. Firstly, desired handle motion is not equivalent to merely reorienting an object manipulator. Further, Felser does not teach, disclose, suggest, allude to, hint or otherwise describe, implicitly or explicitly, a visually confusing orientation or indistinct initial orientation of an object manipulator. To suggest such a teaching extends far beyond the scope of both Wallace and Felser.

In view of the above, Applicants respectfully request allowance of these independent claims.

D. Independent Claims 10, 25, and 40 are Patentable Over Wallace

These independent claims provide for different function states of an object manipulator. Each of the function states enable the object manipulator to perform different discrete functions that modify properties of a graphic object. In other words, a single manipulator can be in various different function states where each state enables a different function. In addition, when the object manipulator is activated, the function state of the manipulator is changed and the property of the graphic object changes. Again, the claims provide that each of the different function states enables a function that modifies a property of the graphic object.

In rejecting these claims, the Office Action relies exclusively on Wallace. The Office Action submits that there are various handles that may be selected. Again, the claims do not recite different manipulators that can be selected with each manipulator available to perform a different function. Instead, each claimed manipulator has multiple different function states that each perform different functions. There is a clear difference between such multiple function states as claimed and the different handles or selection of a single handle in Wallace. The Action also attempts to assert that the selection of the single handset that changes the frame center knob handles is equivalent to the different function states as claimed. Applicants respectfully disagree. Firstly, the frame center knob

handle that has not been selected cannot perform any function that modifies a property of the graphic object Thus, the only function that is performed is that which occurs after the frame center knob handle has been selected. Accordingly, Wallace fails to describe a single object manipulator that has several different function states wherein each function state modifies a property of the object.

The Action then states that there are a discrete number of options that are provided to the user to choose the desired property for manipulation of the graphical object depending on the handle chosen. Applicants again note that while such a fact may be true, the disclosure fails to describe a single handle having different function states that can be changed by activating the manipulator.

The Action further points to FIG. 4 and the ability to drag the frame center knob handle and the different hit points. As claimed, the object manipulator may be in various different function states. When the manipulator is activated, not only does the function state change, but the function associated with the function state and object manipulator changes. Merely dragging a frame center knob handle and determining if it is over a hit point does not even remotely address such different function states or functions as recited in the claims.

In view of the above, Applicants respectfully request allowance of these independent claims.

E. Independent Claims 13, 28, and 43 are Patentable Over Wallace and Argiolas

These independent claims provide for displaying a bitmap image at a cursor position if the cursor position will result in an error when interacting with the object manipulator. In rejecting these claims, the Office Action relies on Wallace and Argiolas. Specifically, to teach the element of the bitmap image for an error condition, the Action relies on Argiolas' teaching that a window size has reached a maximum or minimum size limit.

Applicants submit that Argiolas's field of art is not even remotely similar to that of the present invention. Further, the modification of the size of a window does not teach, disclose, or suggest, implicitly or explicitly, the use of an object manipulator that is displayed on a graphic object in a computer graphics drawing program.

Argiolas' disclosure is directed towards forbidden traffic actions when attempting to resize a window. However, contrary to the implications of the Office Action, a window is not even

remotely similar to a graphic object. In this regard, a window could represent an instance of a drawing program that has graphic objects within it. However, a window and actions performed on the size of a window are not related to nor do they suggest the use of an object manipulator on a graphic object. Further, Applicants note that the error condition as claimed relates to the use of an object manipulator and whether the use of the object manipulator will result in an error condition if the cursor position is selected. However, there are no grips/object manipulators reflected or hinted at in Agriolas. The mere disclosure of a bitmap if a window resizing cannot be performed does not and cannot teach whether the use of a specific object manipulator in a drawing program will result in an error condition.

In addition, there is no suggestion or motivation to use the forbidden traffic action visual signals of Argiolas with the handles set forth in Wallace. Wallace merely describes the use of various handles in a drawing program. However, to use the visual indicators of Argiolas with the specific handles of Wallace extends far beyond the scope of either Wallace or Argiolas.

In fact, even if Argiolas is combined with Wallace, the combination would still fail to teach the invention. For example, the combination would teach Wallace's handles and drawing program within a window that may be resized having visual traffic signals in accordance with Argiolas. However, the combination would not disclose nor suggest the use of Argiolas' traffic signals with Wallace's handles.

In view of the above, Applicants respectfully request allowance of these independent claims.

F. <u>Independent Claims 14, 29, and 44 are Patentable Over Wallace and Young</u>

These independent claims provide provide that a particular glyph shape of the object manipulator indicates an alignment of the object with respect to additional objects. In other words, the claims provide for displaying an object manipulator glyph that is a grip directly on a graphic object. Further, the shape of the glyph indicates an alignment of the object with respect to other objects.

In rejecting these claims, the Office Action relies on both Wallace and Young. More specifically, the Action relies on Young to teach the alignment related aspects of the claims. Applicants respectfully disagree with and traverse the rejections. Young's description relates to displaying guidelines and guidepoints. The Office Action asserts that the guidepoints are equivalent

to Wallace's handles. Assuming such is true, to meet the claim limitations, the guidepoints would have to contain a glyph that indicates an alignment of the graphic object with respect to one or more additional objects. Instead of indicating such an alignment, Young displays a small circle around a point that merely indicates that the point is the guide point associated with the guideline (see col. 6, lines 32-34). However, there is no glyph displayed in Young that indicates an alignment with another graphic object. Further, as explicitly claimed, the shape of Young's circle does not indicate such an alignment. Again, the claims expressly provide that the glyph shape of the object manipulator indicates an alignment of the graphic object with respect to one or more additional objects. A mere circle is not a particular glyph shape that indicates an alignment. Instead, Young's circle merely indicates that a particular point is associated with a guideline. Such a teaching is not remotely similar, nor does it hint or suggest the particular glyph shape and meaning of the glyph shape as claimed.

In view of the above, Applicants respectfully request allowance of these claims.

G. <u>Independent Claims 15, 30, and 45 are Patentable Over Wallace and Frank</u>

These independent claims provide that the object manipulator is displayed in a translucent color such that the graphic object is visible behind the object manipulator. More specifically, like the other claims, the object manipulator is displayed on a graphic object. However, unlike the other claims, the object manipulator itself is translucent such that the graphic object on which it is displayed is visible behind the object manipulator.

In rejecting these claims, the Action relies primarily on Frank. Namely, the Action submits that Frank's buttons that allow the selection of a particular window are equivalent to the claimed object manipulators. The Action then concludes that the windows are transparent and therefore a graphic object can be seen behind a button. However, the Action is misinterpreting the reference. Firstly, it is not the buttons 285-288 that are transparent in Frank. In this regard, the windows are transparent. Further, nowhere in Frank in either FIG. 10, or col. 9, line 60-col. 10, line 25 is there a description of a graphic object that can be seen behind a button. Instead, the buttons may all be seen. In this regard, it is unknown if any graphic object would be visible behind button 282, 286 or 277. Instead, Figure 10 merely shows the ability to see each button with nothing visible behind each button.

The claims explicitly provide that the object manipulator itself is transparent and not the window. Further, the claims also explicitly provide that a graphic object is visible behind the translucent object manipulator. In this regard, the claims do not provide that a window is translucent and a button may be seen through a window. It is completely unknown in Frank if the buttons are translucent or not. Further, Frank fails to describe any such translucency of a button.

In addition to the above, Applicants also submit the Frank merely relates to overlapping windows of a display and is not remotely relevant or related to a computer graphics drawing program as claimed. In view of the above, Applicants respectfully request allowance of these independent claims.

IV. CONCLUSION

In view of the above, Applicants submit that the various elements of Applicants' claimed invention together provide operational advantages over the systems disclosed in Wallace, Choi, Felser, Wang, Argiolas, Young, and Frank. In addition, Applicants' invention solves problems not recognized by Wallace, Choi, Felser, Wang, Argiolas, Young, and Frank.

Thus, Applicants submit that independent claims 1, 7, 8, 10, 13-16, 22, 23, 25, 28-31, 37, 38, 40, and 43-45, are allowable over Wallace, Choi, Felser, Wang, Argiolas, Young, and Frank. Further, dependent claims 3-6, 9, 11, 12, 18-21, 24, 26, 27, 33-36, 39, 41, and 42 are submitted to be allowable over Wallace, Choi, Felser, Wang, Argiolas, Young, and Frank in the same manner, because they are dependent on independent claims 1, 7, 8, 10, 13-16, 22, 23, 25, 28-31, 37, 38, 40, and 43-45, respectively, and because they contain all the limitations of the independent claims. In addition, dependent claims 3-6, 9, 11, 12, 18-21, 24, 26, 27, 33-36, 39, 41, and 42 recite additional novel elements not shown by Wallace, Choi, Felser, Wang, Argiolas, Young, and Frank.

In view of the above, it is submitted that this application is now in good order for allowance and such allowance is respectfully solicited. Should the Examiner believe minor matters still remain that can be resolved in a telephone interview, the Examiner is urged to call Applicants' undersigned attorney.

Respectfully submitted,

GATES & COOPER LLP
Attorneys for Applicant(s)

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: September 12, 2005

By: Jason S. Feldmar
Name: Jason S. Feldmar
Reg. No.: 39,187

JSF/mrj